



Implementing and Evaluating Health as a Service in Fog and Cloud Computing using Raspberry Pi

Pratik Kanani^{1*} Mamta Padole²

¹Dwarkadas J. Sanghvi College of Engineering, University of Mumbai, Maharashtra, India

²The Maharaja Sayajirao University of Baroda, Gujarat, India

* Corresponding author's Email: pratikkanani123@gmail.com

Abstract: Internet of Things (IoT) generates a myriad amount of data, which is sent over the Cloud computing infrastructure for analytics and Business Intelligence. This application scenario suffers network delays, transmission delays and delays in decision making. Due to these drawbacks, the Cloud-based IoT infrastructure is not suitable for time-critical health care applications. To overcome this problem, a smart way is introduced called “Fog Computing” - a LAN based processing approach which has multiple advantages. When IoT, Fog and Cloud Computing are combined, the resultant system's performance is far better. Hence, the combination results in a very efficient Health Care system. Fog and Cloud Computing have their dimensions that not only support each other but also explore many new application domains. In this paper, the real-time ElectroCardioGram (ECG) based Health Care system is implemented in Cloud and Fog Computing. Different Quality of Service (QoS) parameters like memory consumption, transmission delays, computation delays, network delays, Carbon dioxide emission, data transferred and response time are measured, analyzed and improved to make the system more efficient. Based on the Fog computing characteristics and capabilities, the Raspberry Pi 3 B+ model is configured as a Health Care serving gateway by using different installation and configuration steps. Initially, the proposed system is tested for one patients ECG data analysis over cloud and Fog. In every set up all QoS parameters are measured and later the system is subjected to multiple ECG streams for varying numbers of patients to find the limitations of the Raspberry Pi node as a Fog Computing node. The obtained results show that for more number of ECG streams the Fog node is not able maintain QoS in decision making time. Every QoS parameter is explored in detail for decision-making time. In the end, the Fog computing based proposed system is concluded for its pros and cons and future aspects of the Fog node are discussed to make better systems.

Keywords: Cloud computing, Fog computing, Health care system, QoS parameters, Raspberry pi, Real-time ECG analysis.

1. Introduction

In recent years, the dramatic growth of the IoT linearly increases the unprecedented volume and variety of stream data. IoT [1] is a dynamic and global network infrastructure interconnecting objects with unique identities for diverse and advanced application services. Despite offering the advanced services, IoT fails in manipulating the massive amount of data due to its limited storage and processing capacity. Cloud computing technology has unlimited capabilities regarding storage and processing resources, resolving the inconvenience of

IoT by providing virtual resources in a pay-as-you-go manner [2]. Although the vast utilization of cloud computing involves cloud services, resources, applications, several kinds of services and applications, have not completely attained the benefits of the cloud infrastructure due to the latency concern. Owing to the rapid increase of numerous internet-connected smart devices and frequent services, different requests pose a heavy burden on the network bandwidth. The smart devices connected to the cloud server, degrade the QoS. Also, the high network latency between the smart devices and the cloud is infeasible for delay-sensitive applications [3].

Fog computing [4] is the most promising paradigm that significantly reduces the latency and provides the advantages of cloud computing by extending the cloud resources to the network edge [5]. It offers distributed services and allows the knowledge generation and data analytics of the streams generated by the smart IoT devices. The benefits of fog computing are especially useful for pervasive healthcare monitoring applications [6]. The IoT plays a crucial role in continually monitoring the physiological status of hospitalized patients without the need of actively engaging the caregiver [7]. Healthcare monitoring applications [8] widely rely on Wireless Body Area Networks (WBAN), which is the most underlying technology in healthcare IoT. WBAN assists to ubiquitously acquire the physiological information involving Electromyography (EMG), Electrocardiography (ECG), blood pressure, and blood temperature in an efficient and unobtrusive way. To effectively support the pervasive healthcare applications, the prior research works utilize the cloud computing technology for IoT devices [9]. The conventional fog computing methods [10] present a variety of solutions by focusing on the different application scenarios in mitigating the issue of service latency. However, these techniques are still in the nascent stage of attempting to provide the services to the pervasive healthcare computing in the real-world. Thus, the proposed approach focuses on introducing a smart fog gateway by applying the smart partitioning and decision-making using the linear decision tree in fog environment and optimally utilizing the cloud resources for the healthcare IoT requests. It intends to reduce the response latency and increase resource utilization, which is defined in Service Level Agreements (SLAs) while providing the service to healthcare applications.

In this paper, the real-time ECG based analysis system is implemented which analyzes the real-time ECG signal to find abnormalities in it. This analysis is very critical in terms of time and is highly delay-sensitive, as the patient's life can be in grave danger. The ECG based Health Care system is first implemented on Cloud computing and different QoS parameters are studied and recorded. The same system is then migrated to Fog Computing and again the QoS parameters are observed. Different QoS parameters in the context of Health Care are explained and discussed.

To perform the Fog computation Raspberry Pi model 3 B+ is chosen, based on its characteristics. To compare these two architectures, the ECG analysis system is made to run simultaneously on both architectures. The common source of ECG sends the

signal to both the Fog and the Cloud computing nodes. Both nodes do same set of calculations of ECG intervals and analyze the signal in terms of abnormalities. And if the signal is found abnormal then the doctor, hospital staff and patient's relatives are notified by the SMS service. In case of an abnormality in the ECG, if the Short Message Service (SMS) signal reaches early then the chances of the patient getting a faster treatment increases and life can be saved. Fog computing also helps to save energy in terms of power and processing as it supports green computation and is explained in brief in the upcoming sections. In the last section, possible improvements in the Fog computing are discussed such as the capability to run Machine Learning algorithms will lead to drastic improvements in performance and efficiency. Basically, this paper focuses on the use of Fog Computing as the better and more efficient implementation of health care solutions as its use will reduce the Cloud burden in terms of redundant data upload by saving network bandwidth.

1.1 Aim and objectives

- To extend the cloud computing to the fog computing to support latency-sensitive healthcare IoT applications
- To design the smart fog gateway with smart allocation to satisfy the Service Level Agreements (SLAs) in terms of ensuring the optimal response time and resource utilization
- To develop an algorithm to dynamically take a decision regarding IoT stream health care data and decision-making in fog and cloud
- To develop robust and cost and energy effective model in health care to serve the community

1.2 Focus of the paper

- Interactive communication by fog layer even if connectivity to the cloud layer is not present
- Low latency health care application for delay-sensitive real-time applications
- Analysing Fog computing performance with respect to Cloud computing
- Covering Application and Deployment aspects of Fog Computing
- Finding limitation of the Fog computing system for health care

1.3 Paper organization

Section II describes the literature review for the proposed system, followed by the problem statement

and the research gap. Sections V and VI describe the proposed methodology and the Task detailing doing the ECG wave analysis. Further, overall system design and its hardware detailing are discussed along with installation steps. In the end, different QoS parameters are discussed and measured. Different Graphical User Interfaces (GUIs) that are designed for user interactions are shown in the Experimental result section. Finally, the work is concluded and future aspects of the Fog node are discussed.

2. Related work

In recent years, the development and dissemination of smart healthcare systems have received considerable attention from the convergence of a variety of IoT devices. The previous researchers attempted to mitigate the network delay while accessing the centralized cloud environment. The work [11] focuses on improving network performance by grouping the distributed cloud resources into micro data-centers based on network latency, which ensures a tolerable network latency. IoT-based Physical Activity Monitoring (PAMIoT) framework [12] utilizes the cloud services to handle and identify the physical activity information of a human body. It employs IEEE 802.15.4 and Bluetooth technologies to measure the dissolved oxygen in the blood, ECG, and number of steps. The CloudIoT architecture [13] eases the IoT service delivery by introducing the virtual vertical service delivery based IoT PaaS platform. Also, it presents the domain mediation to provide a solution for domain-specific control applications.

Later, fog computing introduces a novel way of moving the cloud infrastructure, proximity to the IoT devices, which provides more opportunities to the IoT [14]. Fog micro data-center [15] dynamically manages the resources for IoT deployments, which is a service-oriented resource management framework. It predicts the utilization of the resources by the users and pre-allocates the resources by exploiting the knowledge of user behaviour and the probability of resource usage in the future. Ubiquitous Data Accessing method (UDA-IoT) in the IoT systems effectively provides the response to the emergency medical services by ubiquitously acquiring and processing the medical data, which improves the accessibility of IoT data resources [16]. Healthcare services mostly depend on the network connectivity and hence, to tackle interruption or delay by the network, the work [17] integrates the IoT with fog computing, offering the cloud resources with effective network performance. Dynamic resource estimation and pricing model [18] co-locates the

smart gateways to create Fog-based micro data-centers for IoT, which ensures efficient and effective resource management in the IoT systems. To resolve the latency issue, the work [19] presents a smart fog gateway with Fog computing in which the architecture performs pre-processing and trimming process before transforming the cloud of things data into the cloud server. The ECG feature extraction model [20] enhances the traditional health monitoring system by smart fog gateway incorporating embedded data mining, notification service, and the distributed fog data storage methods at the edge of the network.

Smart e-health gateway [21] ubiquitously offers local storage, embedded data mining, and real-time local data processing by exploiting the strategic position of gateways, which deals with the scalability, reliability, and energy efficiency issues while considering the burden of sensor networks and remote data centers. iFogSim model [22] extends the cloud services to the edge of the network and decreases the network congestion by enhancing the resource management techniques, which performs real-time analytics and identifies the optimal place of applications on the edge devices. Architecture in [23] presents an IoT service delegation and resource allocation based on the linearized decision tree to diminish the latency for delay-sensitive applications based on collaboration between the fog and cloud environment. An extended work of [21] is presented in [24], in which the smart e-health fog gateway supports the interoperability, reliability, and fog-based mobility support in the healthcare IoT systems. Dynamic Fog model [25] provides a service for time-sensitive healthcare applications involving large-scale, geospatially distributed, and latency-sensitive applications. It analyses the most time-sensitive data of the Heart rate medical data to evaluate the performance of the dynamic fog model. Distributed analytics and edge intelligence model [26] explores the fog computing for pervasive health monitoring applications in terms of real-time fall detection. Based on the Literature review done [11-26], it clears that the Fog computing is meant to process near the source to reduce the cloud burden. Also, it has the capabilities to achieve real time processing and if used in the health care sector then it will be more and more beneficial to the society.

In [56] authors have developed the model where Real time ECG signals are recorded and plotted on the screen. But the main constrain was the internet connection without which the system will not work. In [57], the model ECG based analysis model is proposed which discussed the effect of geographical constraint on the energy efficiency, and it proves that

the fog computing based health care architecture is capable to save the energy. S. Alessandro et al. in article [58] have used fog computing to detect the arrhythmias in the ECG signal. They have used different machine learning techniques for the same. Article [59] says, the Fog computing is a scalable solution to the cloud computing which can store and process near the edge devices.

Articles referred [56-59], are using fog computing approach for health care need. They are also classifying the arrhythmias from ECG signals and processing ECG signals too. But no article is discussing about the Fog Node, its need, configuration parameters, operational setups and deployment issues. Also, Fog these articles say that the Fog node can be used for real time processing and computations, but how fast this computation and at what scale it is better with respect to cloud computing and different QoS parameters are also not discussed.

3. Problem statement

Nowadays, IoT-driven healthcare applications play a vital role in a distributed environment. IoT constantly generates a massive amount of stream data to a greater extent, which leads to the complexity in handling this huge amount of data streams in the IoT devices itself. Since, the IoT devices are resource-constrained devices with respect to the limited storage and processing capability, especially network resources. Moreover, the integrated cloud and IoT technology also impose several challenges for the end-users, network, and the terminals associated with the problem of high congestion, fast battery consumption, and low scalability. Since the long distance between the smart IoT devices and the cloud server creates a gap in providing the response, which leads to the latency issue. Accordingly, the latency issue creates a greater negative impact on healthcare applications due to the fact that healthcare applications are delay-sensitive applications in the real world. Even though the fog computing paradigm provides opportunities to the end-users, the Cloud-Fog interface encompasses several challenges such as context-based resource allocation, workload imbalance, and service overhead. It consumes more time to identify the available Virtual Machines (VMs) from the distributed fog environment to a centralized cloud environment, which degrades the performance of the service when dealing with delay-sensitive healthcare applications. Hence, there is an essential need for satisfying the SLAs in terms, to ensure quick response time and better resource utilization. Moreover, Fog computing does not have the ability to perform the compute-intensive process,

provide massive storage, and establish the wide-area connectivity. Also, dividing the computing of the application in the fog and sending the compute-intensive process to the centralized cloud are the arduous tasks, due to the occurrence of high network latency. Thus, this work targets on providing the smart fog gateway for delay-sensitive healthcare applications by smart partitioning and allocation.

4. Research gap

Most of the formerly presented fog computing research works present the different architectures and frameworks for latency reduction in healthcare IoT systems. However, these methods are not able to effectively utilize the fog as well as cloud environments with the knowledge of application context and resource availability. Nonetheless, the prior architectures fail to analyze the real time ECG signal with lighter computations to provide processing on the go. Moreover, it lacks in faster basic decision making, optimally allocating the application of the edge devices and storing the context-defined fog computing data in the transient fog storage. Define and measure different QoS parameters to measure the Quality of the Fog Computing.

5. Proposed methodology

With the rapid pace of application developments in pervasive and context-aware computing, the Internet of Things (IoT) has become an integral part of day-to-day human life. It is the web working on smart devices embedded with software, electronics, sensors, and network connectivity, ubiquitously providing diverse and advanced services. Accordingly, service discovery, resource management, and energy management require more desirable infrastructure and sophisticated mechanisms. Moreover, standalone energy-constrained IoTs deal with several bottlenecks while continuously generating the stream data. The integration of Cloud computing with the IOTs plays a crucial role in resolving the bottleneck of IoTs. However, cloud-IoT integration involves many challenges for delay-sensitive applications due to the high network latency of the centralized cloud server. Hence, the delay-sensitive applications, especially healthcare applications demand smart gateway such as fog computing to extend the cloud services to the edge of the network. In essence to this, the proposed approach targets to develop the smart fog gateway, involving smart partitioning and allocation by exploiting the decision tree and the dynamic allocation method. It utilizes the knowledge of both the application context and the SLAs to attain

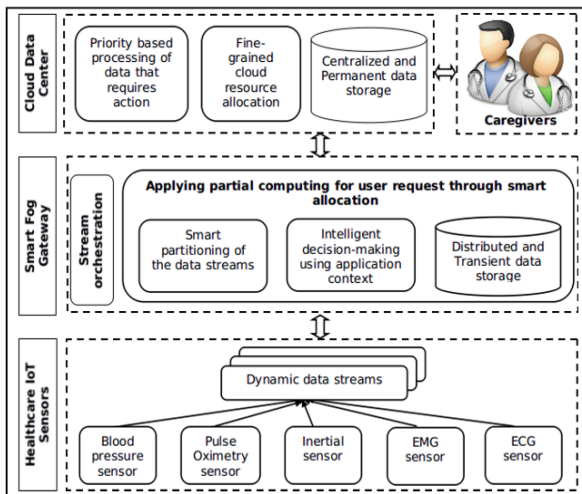


Figure. 1 The process of smart allocation from fog to cloud environment

the QoS. Fig. 1 shows the overall process of the proposed methodology.

In smart fog gateway, the proposed approach applies only to partial computing of user requests by smart allocation. This analysis is done by the optimal placement of the application of the fog resources. In the cloud environment, the proposed approach allocates the partial computing data on the cloud resources using decision rules associated with the parameters of response time, resource availability, and resource utilization while maintaining the priority of actions among the requests received from the distributed fog environment. Finally, the proposed approach analyses the centralized cloud storage according to the execution results and provides the desired service to the end-users. It also focuses on cost [27, 28] and energy-saving solutions for health care.

6. Decision making

Electrocardiogram (ECG) is a graph generated by the electrical activity of the heart muscles. Through ECG one can determine whether the heart is working in normal conditions. Whenever the heart is not working in normal conditions the ECG rhythm changes which is called “Arrhythmia”. ECG helps us to detect abnormal heartbeats, the status of blood supply in the heart due to cholesterol clogging and enlargements in the heart [29].

PR and QT are the main intervals in the ECG wave. A PR wave is generated when the left atrium receives an electrical impulse from the right atrium. The QRS complex gets generated when both ventricles begin to pump, and at this time, a “beep” sound is generated in the cardiac monitors. When the initial contraction is over it results in ST-segment

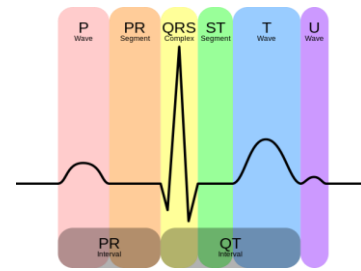


Figure. 2 Complete ECG wave with intervals

generation. And when ventricles are relaxing, the T wave is generated. The normal beats are from 60-100 with these intervals present in the ECG waveform [30]. These intervals are shown in Fig. 2 [31]. After receiving the ECG signal, we can use different algorithms and approaches like wavelet transformations, wavelet analysis, Pen and Tomkins algorithm, template-based matching, QRS peak detection, neural networks or windowing algorithms to process and analyze the same.

ECG signals can be analyzed in frequency or in the time domain. The Frequency domain algorithm needs complex mathematical calculations and different transformations, while in the time domain by using a simple windowing algorithm with precise calculations the ECG wave intervals can be found out. In this paper, the Novel windowing algorithm developed by Muhammad et al. in [32] is used and is modified partially to analyze the ECG wave.

In this system, the live ECG signal is captured [33] using three AD8232 electrode-based sensor and Analog to Digital Converter (ADC) is done by Arduino Nano.

Arduino is a small micro-controller capable of doing many things. It has multiple analog pins. And in the Arduino library, we have functions like Analog Read (), which will read the analog input from the particular pin of Arduino and will convert it into a digital value ranging in 0-1023. It has a 10-bit binary resolution. Here the ECG readings obtained in real-time using the AD8232 sensors are in the range of 0-1023. These readings when plotted show the actual ECG wave. In this, the sampling frequency (f_s) used is 500 Hz, which is between two points the time interval is 2 ms. The recorded ECG signal is plotted in Fig. 3 and it is in the range of 0-1023.

For the proposed system, the obtained real-time ECG records have an R peak above 600 thresholds. The ECG readings are stored in the moving array where each index represents one respective voltage value and the time gap between the adjacent indices is 2 ms.

ECG is a periodic wave, it repeats its cycles after a certain interval of time. Now the important part is to get PR, QRS and QT intervals out of these waves

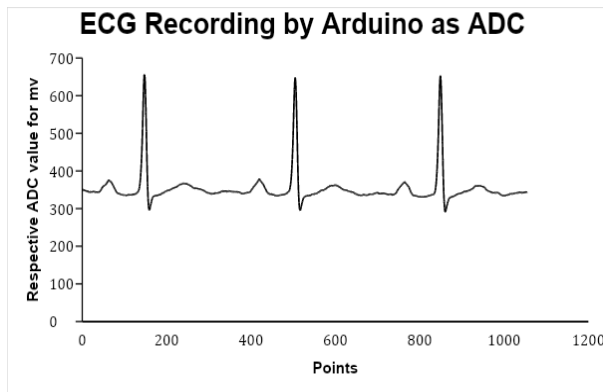


Figure. 3 ECG wave as discrete signal recorded with AD8232 and after ADC by arduino

Table 1. Standard ECG Intervals for a healthy adult with standard bpm of 60

Intervals	Normal Value	Normal Variation
QT Intervals	400 ms	±40ms
QRS Interval	100ms	±20ms
PR Interval	160 ms	±40ms

for each and every wave.

Based on the given table we can find the time intervals in milliseconds and by comparing with table 2 [34-37] we can find whether ECG waves are normal or abnormal. The normal beats per minute (bpm) are 60 to 100 bpm. Further, we have used the windowing algorithm to detect different intervals.

All real-time readings are stored in the array and the intervals are found by using the following algorithm.

Algorithm 1 Windowing Algorithm

Input: ECG wave, **Output:** Ω

// $\Omega \in \{\text{normal, abnormal}\}$

// i is each ECG wave ranging from P-T-R

1. **procedure** detect
2. **for** i from 1 to n **do**
3. R-R interval is $t_{rr} = \frac{R_{(i+1)} - R_{(i)}}{f_s}$
4. P-R interval is $t_{pr} = \frac{R_{(i)} - P_{(i)}}{f_s}$
5. QRS interval is $t_{qrs} = \frac{(S_{(i)+8}) - (Q_{(i)-8})}{f_s}$
6. QT interval is $t_{qt} = \frac{T_{(i)} + (t_{rr} * 0.13) - (Q_{(i)-8})}{f_s}$
7. $bpm = \frac{t_{rr} * 60}{f_s}$
8. **End for**
9. **End procedure**

7. System architecture

In the proposed methodology, a real-time health care system is taken into consideration. The system consists of a Data source, Fog Node, Gateway, Decision making, and messaging service. The real-time ECG signal is acquired from the patient [33].

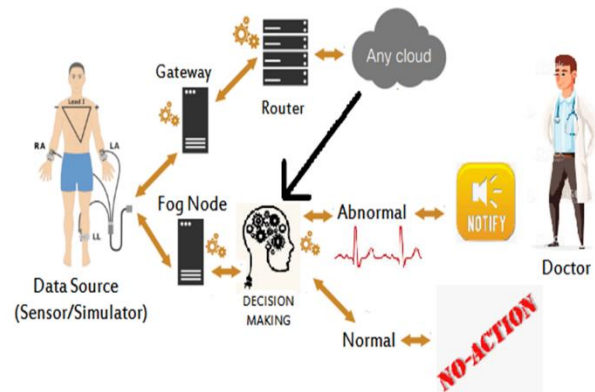


Figure. 4 Cloud computing and fog computing based health care system

These signals are recorded and sent to Cloud as well as the Fog Computing node. The windowing algorithm [32] is used to find the reference points PQRST in the ECG Signal. Based on these points the ECG time intervals are found out. Later decision making is done to find whether the given ECG signal is normal or abnormal. In the proposed architecture shown in Fig. 4, the real-time ECG signals are sent to the Cloud and Fog node simultaneously.

The task of performing analysis is carried out on both the systems on the same signal. The generated results are then compared in terms of different parameters like computation time, transmission time, Carbon dioxide (CO₂) generated and the total response time. Whenever an abnormality in the signal is found, at that very instant a text message consisting of the timestamp, signal interval values, and the patient data, is sent to the doctor and based on the timestamp the response efficiency is calculated.

8. Raspberry pi as the fog computing node and its environmental setup

Raspberry Pi Model 3 B+ is used and in order to make it a Fog Computing node, different environmental installation and setups are done on it. The steps are explained below.

8.1 Setup of NOOBS operating system on raspberry pi

Here the NOOBS Operating System [38-39] is used i.e. “New out of Box Software” operating system. Below are the installation steps of the NOOBS Operating system On Raspberry Pi.

Step 1: Download NOOBS and extract it.

Step 2: Format an SD card with the use of SD Association’s Formatting Tool.

While formatting makes sure that the “Format size adjustment” option is set to “on.” So that the developer can erase it in FAT format.

Step 3: Now one has to just drag and drop the NOOBS files into your newly formatted SD card.

Step 4: Put the SD card into Raspberry Pi and boot it up.

Step 5: Once the SD card has booted noobs, it starts running the setup at 12V on 1 AMP power supply.

Step 6: Once the noobs OS is set up, one gets a customized Linux desktop with user-id “pi” and password “raspberrypi”.

Step 7: The next step is to setup the apache tomcat server for web application deployment and MySQL for local database storage.

8.2 Setup of apache tomcat server

Here, “sudo” command is used for super users. Below are the steps for the setup of the Tomcat server [40].

Step 1: `sudo apt-get update`

Step 2: `sudo apt-get upgrade`

For installing MySQL put the following commands

Step 3: `sudo apt-get install mysql-server --fix-missing`

Step 4: `sudo apt-get install mysql-workbench`

8.3 For installing tomcat server run the following command

Step 1: `sudo apt-get install tomcat8`

Step 2: `sudo apt-get install tomcat8-admin`

After the installation of tomcat8 for tomcat8-admin, put both username and password e.g., manager for deploying war file. Open the tomcat manager page using `http://RASPBerry_IP:8080/manager/html`

8.4 Generate and deploy war file

To create a war (Web Archive) file of the project, first, go inside the project directory. In the project folder

1. From your host machine, Right Click on Dynamic web project name in Eclipse
2. Select Export Option
3. Save as war file
4. Open the tomcat manager page on raspberry
5. Import the war file on the manager page and use the upload war option. Save the war file and use it for both Fog computing and Cloud Computing.

9. QoS parameters

Different QoS parameters like memory [41-42], Transmission delay, Computation delay, CO₂ emission measurement [43-46], data transferred and Response time are as follows.

1) Memory

The amount of memory utilized by a programming module can be calculated using Java methods. One can use the Runtime class functions for finding the memory utilized.

$$\begin{aligned} \text{Long usedMemory} = \\ \text{Runtime.getRuntime().totalMemory()} - \\ \text{Runtime.getRuntime().freeMemory()} \quad (1) \end{aligned}$$

The memory used for the ECG analysis is calculated as the difference of total available memory and the free memory.

2) Transmission Delay

Transmission delay is the total time taken by the network to send the data from one point (source) to the second point (Destination).

3) Computation Delay

Computation Delay is the total time taken for the computation. In the proposed system it is the time measure after the entire signal is received in the system until it is processed and the output as normal or abnormal is produced. Based on the computation time the Speed up in the Fog Computing can be calculated by

$$\text{Speedup}_{\text{overall}} = \frac{\text{Execution time}_{\text{cloud}}}{\text{Execution time}_{\text{fog}}} \quad (2)$$

Speedup helps to determine which computation is faster and how much. Here the overall speedup is showing how faster the fog decisions are with respect to the cloud computing. Here, the execution time is representing the overall execution time of Cloud and the Fog node respectively in numerator and in the denominator.

4) CO₂ Measurement

Fog Computing uses far fewer resources than the Cloud Computing infrastructures which reduce the CO₂ generation. The amount of CO₂ that can be saved from Fog Computing is further explained below.

In 2011, According to Cisco 1.8 ZB of data was sent to the Cloud data centers. So, if 5.12 kWh of energy is required to send 1 GB data across then the total energy required to send 1.8 trillion GB of data is 9.216 trillion kWh of energy. To generate this much amount of energy, a total of 5.76 trillion kg of CO₂ is emitted. Using the above inferences, to transfer 1 Byte of data 2.98 x 10⁻³mgm of CO₂ is emitted.

Now, if one can adapt to Fog Computing and assume that the data only travels to the data center only for storage purposes, then one reduces the CO₂ emission by 50%, and the CO₂ emission can be potentially reduced by 2.88 billion metric tons. That is almost 34 times more savings when compared to Cloud Computing. The amount of CO₂ can also be found out by finding the power used by the devices and relating CO₂ with power.

$$Power_{static} = Current_{static} \times Voltage \quad (3)$$

$$Power_{dynamic} = \frac{1}{2} \times Capacitive\ load \times Voltage^2 \times Frequency\ switched \quad (4)$$

Static power is the power consumed by the device where no input is active and the dynamic power is the power consumed by the device when the input signal is active and output signal is getting processed/generated. The power consumption of any device is proportional to the amount of CO₂ generated indirectly. Here, current and the voltage are representing the input current and the voltage need of the device. Capacitive load is the capacitance of the wires and transistors used. And the frequency switched is the clock rate used.

5) Data Transferred

The data transferred is measured in bytes. Here, the ECG signals are sent to the Cloud and the Fog node. The network distance is measured in terms of the number of Hops. Usually, the Fog Computing node is at least 2 to 3 hops away from the data source.

6) Response Time

For the proposed system architecture, the Response time considered is the overall response time of the ECG signal processing unit. Here it shows the total time span starting from the ECG signal generation until the final response is generated and

given to the doctor. It includes the addition of other delay times like Processing Delay, Queuing Delay, Transmission Delay, and the Propagation Delay.

$$d_{response} = d_{proc} + d_{queue} + d_{trans} + d_{prop} \quad (5)$$

The final response time to get the output is the sum of processing, queueing, transmission and propagation delays respectively.

For the current system, the timestamp value of the Java programming language counts as the current system time in milliseconds which is noted by using the java method

$$System.\ currentTimeMillis(); \quad (6)$$

This function represents the current system clock time in milliseconds. It is a long data type number. Here, the final system improvement of the Fog computing is shown by the difference of Response time taken by both the systems.

9.1 Different time stamps and defining the QoS parameters

In this system, different parameters are defined to measure the QoS parameters, and arithmetic relations among these parameters are used to calculate different delays. The timestamp Parameters with their denotations are as follows.

10. Experimental results

The web interface is developed to understand the ECG signal processing and its analysis in more detail. The same web interface is run on both Cloud and the Fog Computing node. The interface shows the

Table 2. Different time stamp calculations

Different instants	Timestamp	QoS Parameters
T_{gen} = ECG signal generation time		Overall Response Time improved by Fog = $T_{pc} - T_{pf}$
T_{rf} = Time at which ECG signal is reaching the Fog Node		Transmission Delay (Fog) = $T_{rf} - T_{gen}$
T_{pf} = Time at which ECG signal is processed at Fog Node		Transmission Delay (Cloud) = $T_{rc} - T_{gen}$
T_{rc} = Time at which ECG signal reaches the Cloud Computing		Fog Computation Time = $T_{pf} - T_{rf}$
T_{pc} = Time at which ECG signal is processed by the Cloud Computing		Cloud Computation Time = $T_{pc} - T_{rc}$
		End to end Data Transferred = Data bytes x no. of Hops
		CO ₂ generated = Data Transferred × emitted CO ₂ /Byte

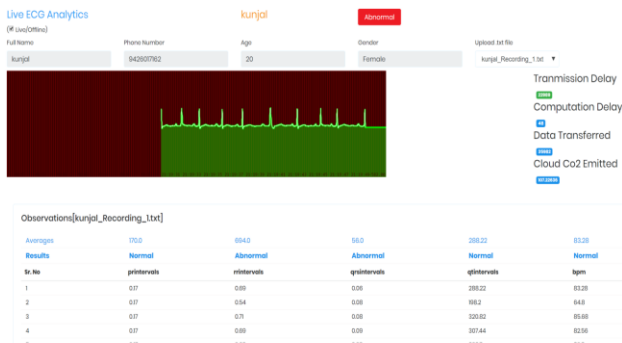


Figure. 5 ECG analyzing system interface on cloud computing

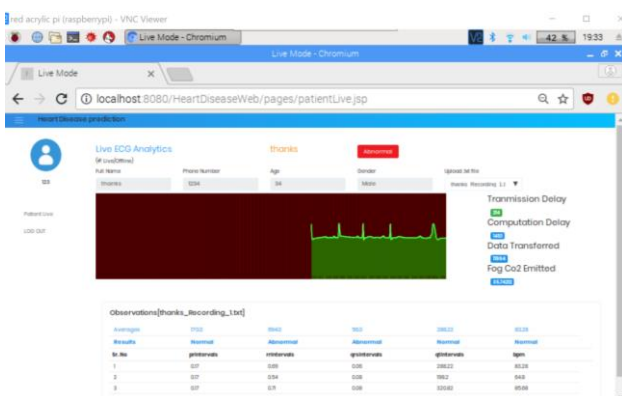


Figure. 6 ECG analyzing system interface on raspberry pi as a fog computing node

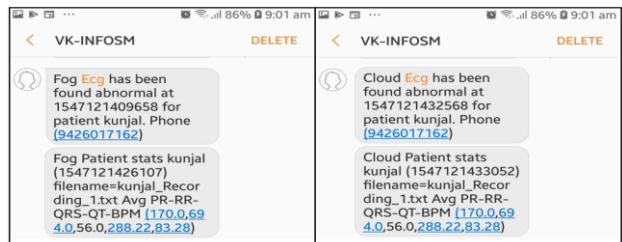


Figure. 7 SMS responses for the fog computing and the cloud computing

Table 3. QoS parameters and their values

Parameters	Cloud Computing	Fog Computing	Improvement
Transmission Delay (ms)	7677	117	7560
Computational Delay (ms)	55	670	-615
Data Transferred (bytes)	35982	11994	23888
CO2 Emitted (mgm)	107.22	35.74	71.48
Response Time in MS (Time Format in Java)	1547121433052	1547121426107	6945

“Patient Name”, a red color label if signal Abnormality exists, the “.txt” file name where the ECG signals are stored for processing, different QoS parameters, ECG waveform, patient details, different intervals for each ECG wave along with their abnormality. If the signal is found abnormal then the system will send SMS to the Health Care supporting staff. All historic “.txt” files are also stored on both the nodes for future referral. The Cloud infrastructure chosen for the system is EverData.com.

The same web interface is deployed on Raspberry Pi 3 b+ node which is acting as a Fog Computing Node. The difference one can see is, it is running on the localhost at the Gateway. The final responses of both the systems are conveyed by the SMSs, where each message contains the computing node from where the message has come, patient name, mobile number, the timestamp at which the message is generated, file name where the signals are stored and the respective ECG intervals which doctor and hospital staff can refer for advance preparations before patient arrival. The network delays from the service provider can affect the receiving time of the SMS, which may cause a delay in response.

When the millisecond timestamps shown in Fig. 7 are Converted [47] actual times, they are 5.27.06 and 5.27.13 for Fog and Cloud respectively. This shows that the Fog node is responding almost six seconds before the Cloud node which is very vital to save the patients when they are in critical condition.

The Fog and Cloud computing system is first subjected only for 1 patient, and all resultant parameters are measured and shown in Table 3. The system shows that Fog computing surpasses overall response time performance than Cloud computing and performs better in terms of Response time, data transferred and CO2 generated. But Fog computing is hanging back in terms of computation power. Cloud computing takes only 55 ms to computer the given job while the Fog processor takes 670 ms.

The system is tested against by varying the number of patients to study the system behaviour in depth. Each and every parameter value is taken and shown as an average value for n patients to discuss further.

10.1 Transmission delay

Transmission Delay depends on many factors like the number of hops between the source and the destination, available network bandwidth, layer conversion, VPN set up, wired-wireless configurations, congestions, tunnelling, and the number of users, etc. A hop occurs when a packet is passed from one network segment to the next. Data

Table 4. Cloud server distances in number of hops

Cloud Servers	Number of Hops
Amazon Web Services	20
Microsoft Azure	12
Google Cloud Platform	09
IBM Cloud	06
Verizon Cloud	11
ThingSpeak	>30
EverData	11

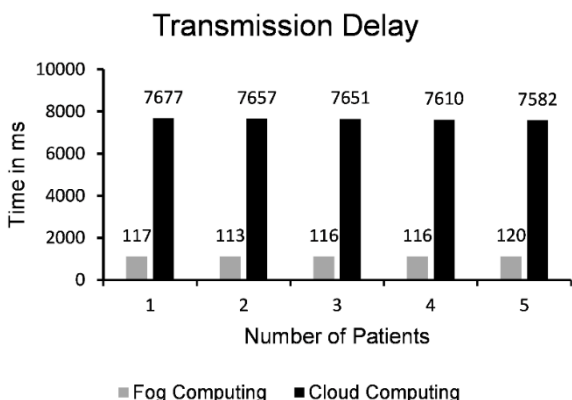


Figure. 8 Transmission delays of fog and cloud computing

packets pass through routers as they travel between source and destination [48]. As WANs tend to have more number of devices connected to it and its huge network covering geographic areas, there is a slight drop in network bandwidth as the time taken to deliver the packet from source to destination has increased. So as number of hops increases, the chances of bandwidth decreasing increases.

$$Bandwidth = \frac{Number\ of\ bits}{Time\ taken} \text{ bits}/sec \quad (7)$$

The bandwidth is defined as the ration of number of bits transferred and the time taken to transfer those bits.

The number of hops in Fog computing is one, but in case of Cloud computing, the number varies as it is in the WAN. Different Cloud servers are tested for their distance by the number of hops and by using “tracert” command [49-51]. It is shown in Table 4. Of course, the number of hops count varies from place to place. From Table 4, it is clear that the cloud servers take more hops to reach. Hence by increasing transmission delays. The proposed system is tested for a different number of patients from 1-5. And the average transmission time is shown in Fig. 8.

Fig. 8 makes it clear that the transmission delay in terms of Cloud and Fog Computing remains almost the same for the same source and destination for a different number of patients. And it is also evident

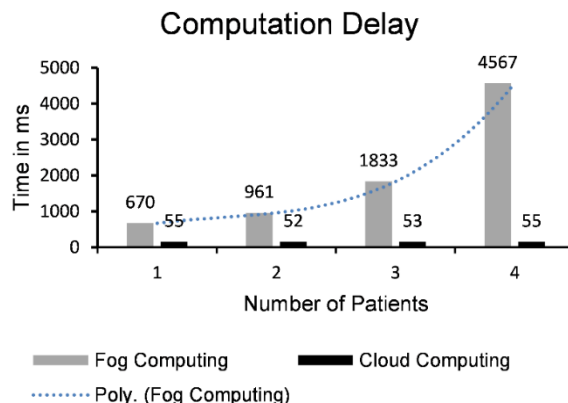


Figure. 9 Computation delay of fog and cloud computing

that the transmission delay in Cloud computing is very much higher than the Fog computing.

10.2 Computation elay

Computing power depends on the device capability in terms of its hardware configuration. It depends on cache memory, processor, operating frequency, scheduling algorithm, communication bus, memory and number of cores. The current Fog device is configured to read and analyse more real-time ECG waves simultaneously on its different ports. By varying the number of patients its average computational delay in fog node is measured and shown in Fig. 9.

The computation delay is almost identical in terms of Cloud computing but it varies a lot in the case of Fog computing. Fog computing shows the polynomial growth of the order of 3°. The computational delay also depends on how the system is made i.e. the GUI computation, background computation, the refresh rate and the number of parallel tasks, etc. For four patients the computational delays are 4523, 4662, 4487 and 4593 ms respectively, which comes out as 4567 as an average value shown in Fig. 9. So giving more load on the Fog system is not suitable for the time-sensitive decision-making systems.

10.3 Response time

The Response time is the overall performance time of the system. It shows the time difference between the generation of the ECG signal and the generation of response or decision in terms of normality and abnormality. Here, the average response time is found by varying the number of patients as shown in Fig. 14. In the proposed system, the Fog node serves early responses than the cloud architecture, but only when the number of users is

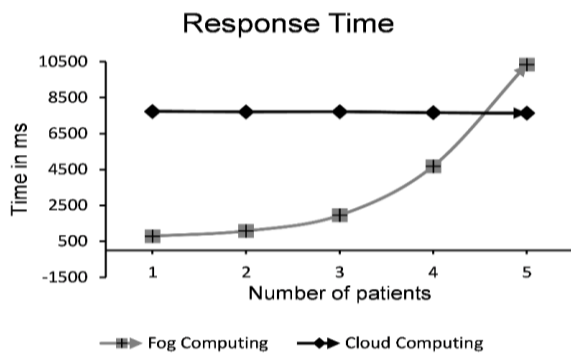


Figure. 10 Response time of fog and cloud computing

lesser than four. It responds very early if the number of users is less. While in case of Cloud computing the Response time is almost equal. If the number of users is five or more than that then the proposed Fog computing architecture will underperform than the cloud.

Hence it is not suggested for more number of patients in real-time health care analytics.

10.4 Standard deviation

Standard deviation shows the spread of numbers from the mean or the expected value. When standard deviation is calculated for the different factors like transmission time and computation time for the Fog and Cloud computing, the Fog processing time shows maximum standard deviation. This deviation is due to varying number of patients in the fog node. It is shown in Fig. 11.

11. Conclusion

This work introduced a cloud-assisted smart fog gateway for delay-sensitive IoT-driven healthcare applications. It ensures the tolerable delay while providing the service to the healthcare applications, by applying the smart partitioning and allocation using a decision tree. The decision rules are based on the application context and resource availability in fog and cloud infrastructure. In the smart fog gateway, the proposed approach intelligently takes the decision to determine the corresponding data stream based on the application context. The proposed approach provides the service to the end-user promptly. Cloud Computing Based IoT architecture is delay-sensitive for Critical Health Care applications. So, the LAN based Fog computing Processing approach can be used to reduce the delay. Also, this technique helps to reduce the data burden on the Cloud. Moreover, Fog Computing should have memory, processing and computation capabilities and Fog nodes can be placed in either LAN or as a Gateway. Since Raspberry Pi

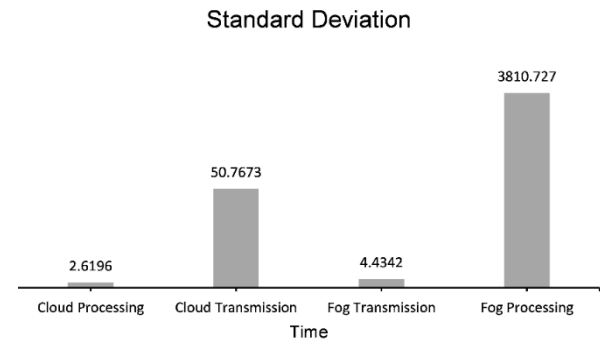


Figure. 11 Standard deviation of different parameters

has networking, memory, storage and computation abilities, it becomes a suitable option to use as a Fog node. We also discuss different Raspberry Pi based Fog installations. Furthermore, fog based health care systems are better than the Cloud-based health care system in terms of network bandwidth and response time, but it lags behind in computation power. The overall response by fog to find any abnormality in the ECG signal is given way before the Cloud does - which is very vital in health care scenarios to save patient's lives. The Windowing algorithm is suitable to classify ECG signal in real time. Fog computing gives a better response when the number of patients is less and it is observed that the Fog computing response time is directly proportional to the number of patients. After a particular threshold value of the number of patients, fog computing will not perform better than Cloud computing. The transmission delay and the computation delay plays a major role in the Fog computing domain. Further, the Fog node can be improved in terms of computations by designing light computation-intensive tasks and it would result in new Fog Computing eras.

12. Fog computing node in future

Raspberry Pi is fairly efficient due to the availability of the hardware and the software at a reasonably low cost. The problem in the current available Raspberry-Pi is the capacity in terms of hardware and main memory. So, because of these limitations one cannot deploy Machine Learning based applications on the Raspberry Pi. Such deployments will make the responses slow, which cannot be tolerated in applications like health care.

Other high-end computation boards are available like Latte Panda alpha, Rock Pi 4, Tinker Board and Beaglebone Black [52-55]. With an increase in the hardware capabilities of these boards, one can now add Machine Learning functionalities to the health care applications without any delays, which will result in the QoS in health care applications. ECG

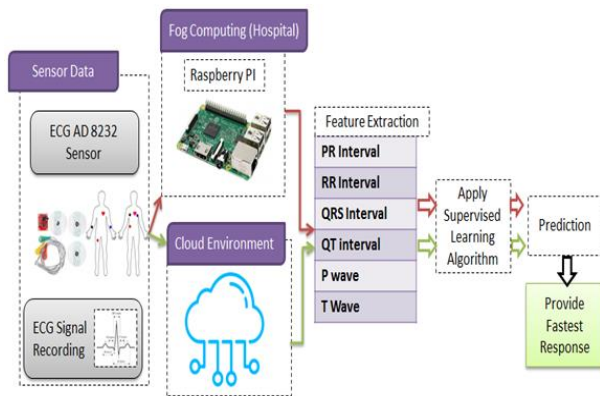


Figure. 12 Machine learning enabled ECG health care application

based Health Care system with Machine Learning capabilities is presented in Fig. 12.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

“Conceptualization and Methodology, Pratik Kanani and Dr. Mamta Padole; writing—original draft preparation, Pratik Kanani; writing—review and editing, Dr. Mamta Padole; visualization, Pratik Kanani; supervision, Dr. Mamta Padole; validation, Pratik Kanani and Dr. Mamta Padole.

References

- [1] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey”, *Elsevier Computer Networks*, Vol. 54, No. 15, pp. 2787-2805, 2010.
- [2] A. Botta, W. De Donato, V. Persico, and A. Pescapé, “Integration of cloud computing and internet of things: a survey”, *Elsevier Future Generation Computer Systems*, Vol. 56, pp. 684-700, 2016.
- [3] M. Hassanlieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, and S. Andreescu, “Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges”, In: *Proc. of IEEE International Conf. on Services Computing (SCC)*, NY, USA, pp. 285-292, 2015.
- [4] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things”, In: *ACM Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, pp. 13-16, 2012.
- [5] H. Chang, A. Hari, S. Mukherjee, and T. V. Lakshman, “Bringing the cloud to the edge”, In: *Proc. of IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, Canada, pp. 346-351, 2014.
- [6] Y. Shi, G. Ding, H. Wang, H. E. Roman and S. Lu, “The fog computing service for healthcare”, *IEEE 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech)*, Beijing, China, pp. 1-5, 2015.
- [7] S. R. Islam, D. Kwak, M. H Kabir, M. Hossain, and K. S. Kwak, “The internet of things for health care: a comprehensive survey”, *IEEE Access*, Vol. 3, pp. 678-708, 2015.
- [8] N. Bui and M. Zorzi, “Health care applications: a solution based on the internet of things”, In: *ACM Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, Barcelona Spain, p. 131, 2011.
- [9] C. Doukas and I. Maglogiannis, “Bringing IoT and cloud computing towards pervasive healthcare”, In: *Proc. of IEEE Sixth International Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, Italy, pp. 922-926, 2012.
- [10] S. Yi, C. Li, and Q. Li, “A survey of fog computing: concepts, applications and issues”, In: *ACM Proceedings of the Workshop on Mobile Big Data*, pp. 37-42, 2015.
- [11] S. Malik, F. Huet, and D. Caromel, “Latency based group discovery algorithm for network aware cloud scheduling”, *Elsevier Future Generation Computer Systems*, Vol. 31, pp. 28-39, 2014.
- [12] P. P. Ray, “Internet of things based physical activity monitoring (PAMIoT): an architectural framework to monitor human physical activity”, In: *Proc. of IEEE CALCON*, pp. 32-34, 2014.
- [13] F. Li, M. Vögler, M. Claeßens, and S. Dustdar, “Efficient and scalable IoT service delivery on cloud”, In: *Proc. of IEEE Sixth International Conf. on Cloud Computing (CLOUD)*, USA, pp. 740-747, 2013.
- [14] M. Chiang and T. Zhang, “Fog and IoT: An overview of research opportunities”, *IEEE Internet of Things Journal*, Vol. 3, No. 6, pp. 854-864, 2016.
- [15] M. Aazam, and E. N. Huh, “Dynamiac resource provisioning through fog micro datacenter”, In: *Proc. of the 12th IEEE International Workshop on Managing Ubiquitous Communication and Services (MUCS '15)*, USA, pp. 105-110, 2015.
- [16] B. Xu, L. Da Xu, H. Cai, C. Xie, J. Hu, and F. Bu, “Ubiquitous data accessing method in IoT-based information system for emergency

- medical services”, *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 2, pp. 1578-1586, 2014.
- [17] F. Andriopoulou, T. Dagiuklas, and T. Orphanoudakis, “Integrating IoT and Fog Computing for Healthcare Service Delivery”, *Springer Components and Services for IoT Platforms*, pp. 213-232, 2017.
- [18] M. Aazam and E. N. Huh, “Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT”, In: *Proc. of IEEE 29th International Conf. on Advanced Information Networking and Applications (AINA)*, Gwangju, Korea, pp. 687-694, 2015.
- [19] M. Aazam and E. N. Huh, “Fog computing and smart gateway based communication for cloud of things”, In: *Proc. of IEEE International Conf. on Future Internet of Things and Cloud (FiCloud)*, Barcelona, Spain, pp. 464-470, 2014.
- [20] T. N. Gia, M. Jiang, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, “Fog computing in healthcare internet of things: A case study on ecg feature extraction”, In: *Proc. of IEEE International Conf. on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*, Liverpool, UK, pp. 356-363, 2015.
- [21] A. M. Rahmani, N. K. Thanigaivelan, T. N. Gia, J. Granados, B. Negash, P. Liljeberg, and H. Tenhunen, “Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems”, In: *Proc. of 12th Annual IEEE Consumer Communications and Networking Conf. (CCNC)*, Las Vegas, NV, USA, pp. 826-834, 2015.
- [22] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, “iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things”, *Edge and Fog Computing Environments*, *arXiv preprint arXiv:1606.02007*, 2016.
- [23] A. A. Alsaffar, H. P. Pham, C. S. Hong, E. N. Huh, and M. Aazam, “An Architecture of IoT Service Delegation and Resource Allocation Based on Collaboration between Fog and Cloud Computing”, *Hindawi Mobile Information Systems*, pp. 1-15, 2016.
- [24] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, “Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach”, Elsevier Future Generation Computer Systems, 2017.
- [25] S. Chakraborty, S. Bhowmick, P. Talaga, and D. P. Agrawal, “Fog Networks in Healthcare Application”, In: *Proc. of IEEE 13th International Conf. on Mobile Ad Hoc and Sensor Systems (MASS)*, Brasilia, Brazil, pp. 386-387, 2016.
- [26] Y. Cao, P. Hou, D. Brown, J. Wang, and S. Chen, “Distributed analytics and edge intelligence: Pervasive health monitoring at the era of fog computing”, In: *ACM Proc. of the Workshop on Mobile Big Data*, pp. 43-48, 2015.
- [27] P. G. Katona and D. Jaron, “Cost Effective Health Care Technologies: A Joint Grant Program of the Nsf and the Whitaker Foundation”, *Health Care Technology Policy I: The Role of Technology in the Cost of Health Care*, Arlington, VA, USA, 1994, pp. 51-54.
- [28] E. Kwak, G. Kim, and J. Yoo, “Network operation cost model to achieve efficient operation and improving cost competitiveness”, In: *Proc. of the 13th International Conf. on Advanced Communication Technology (ICACT2011)*, Seoul, pp. 1107-1112, 2011
- [29] Electrocardiogram (ECG). (2017). BLOOD PRESSURE UK [Online]. Available: <http://www.bloodpressureuk.org/BloodPressureandyou/Medicaltests/ECG>.
- [30] Standard range of intervals. (2017 June, 03). E MEDICINE [Online]. Available: <http://emedicine.medscape.com/article/2172196-overview>.
- [31] PQRST INTERVAL. (2017). SPARK FUN [Online]. Available: https://cdn.sparkfun.com/assets/learn_tutorials/2/5/0/EKG_Complex_en.svg.png.
- [32] Muhammadd U. Bilal Ahmed B et al Muhammad Umer, Bilal Ahmed Bhatti, Muhammad Hammad Tariq, Muhammad Ziaul-Hassan, Muhammad Yaqub Khan, Tahir Zaidi, Feature Extraction and Pattern Recognition Using a Novel windowing Algorithm”, *Advances in Bioscience and Biotechnology*, Vol. 5, pp. 886-894, 2014.
- [33] P. Kanani, M. Padole. (2018) Recognizing Real Time ECG Anomalies Using Arduino, AD8232 and Java. In: Singh M., Gupta P., Tyagi V., Flusser J., Ören T. (eds) *Advances in Computing and Data Sciences. ICACDS 2018. Communications in Computer and Information Science*, Vol. 905. Springer, Singapore
- [34] Cardiology Teaching Package. http://www.nottingham.ac.uk/nursing/practice/resources/cardiology/function/normal_duration.php

- [35] Standard range of intervals, 2017. E MEDICINE. <http://emedicine.medscape.com/article/2172196-overview>
- [36] Normal ECG. https://meds.queensu.ca/central/assets/modules/ECG/normal_ecg.html
- [37] Eduardo José da S. Luz, William Robson Schwartz, Guillermo Cámara-Chávez, and David Menotti “ECG-based heartbeat classification for arrhythmia detection: A survey”, *Computer Methods and Programs in Biomedicine*, Vol. 127, pp. 144-164, 2016.
- [38] NOOBS, <https://www.raspberrypi.org/downloads/noobs/>
- [39] Getting started with NOOBS, <https://www.raspberrypi.org/help/noobs-setup/2/>
- [40] Installing Tomcat8 on Raspberry pi 3, <http://androidsrc.net/installing-tomcat8-raspberry-pi-3/>
- [41] Computer - Memory, https://www.tutorialspoint.com/computer_fundamentals/computer_memory
- [42] How to calculate a memory usage of a java program?, <https://stackoverflow.com/questions/37916136/how-to-calculate-memory-usage-of-a-java-program>
- [43] Cloud Computing Saves Energy and CO₂ Emissions, <http://www.energydigital.com/sustainability/cloud-computing-saves-energy-and-co2-emissions>
- [44] How is cloud influencing world data traffic?, <https://www.ibm.com/blogs/cloud-computing/2013/04/how-is-cloud-influencing-world-data-traffic/>
- [45] The Megawatts behind Your Megabytes: Going from Data-Center to Desktop, <http://aceee.org/files/proceedings/2012/data/papers/0193-000409.pdf>
- [46] With Internet of Things and Big Data, 92% Of Everything We Do Will Be In The Cloud, <https://www.forbes.com/sites/joemckendrick/2016/11/13/with-internet-of-things-and-big-data-92-of-everything-we-do-will-be-in-the-cloud/#553568ed4ed5>
- [47] Current Millis, <https://currentmillis.com/>
- [48] D. Comer and D. L. Stevens. *Internetworking with TCP/IP: Principles, protocols, and architecture*. Internetworking with TCP/IP. Pearson Prentice Hall, 2006. ISBN: 9780131876712. url: <https://books.google.co.in/books?id=jonyuTASbWAC>.
- [49] WHAT CAUSES THE INTERNET TO SLOW DOWN? <https://www.colocationamerica.com/data-center-connectivity/speed-test.htm>
- [50] A. Faggiani, E. Gregori, A. Improta, L. Lenzini, V. Luconi and L. Sani, “A study on traceroute potentiality in revealing the Internet AS-level topology”, In: *Proc. of 2014 IFIP Networking Conference*, Trondheim, pp. 1-9, 2014
- [51] S. Branigan, H. Burch, B. Cheswick, and F. Wojcik, “What can you do with Traceroute?”, *IEEE Internet Computing*, Vol. 5, No. 5, p. 96. 2001.
- [52] Raspberry Pi, <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>
- [53] LattePanda Alpha 864s, <https://www.dfrobot.com/product-1727.html>
- [54] Asus Tinker Board Vs Raspberry Pi 3 – Specifications and Pricing, <https://mybroadband.co.za/news/hardware/195442-asus-tinker-board-vs-raspberry-pi-3-specifications-and-pricing.html>
- [55] Meet BeagleBone Black wireless, <https://beagleboard.org/blog/2016-09-26-meet-beaglebone-black-wireless>
- [56] P. Patil and K. Bhole, “Real time ECG on internet using Raspberry Pi”, In: *Proc. of 2018 International Conf. on Communication, Computing and Internet of Things (IC3IoT)*, Chennai, India, pp. 267-270, 2018, doi: 10.1109/IC3IoT.2018.8668157.
- [57] S. M. Isa, M. O. I. Musa, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Energy Efficient and Resilient Infrastructure for Fog Computing Health Monitoring Applications”, In: *Proc. of 2019 21st International Conf. on Transparent Optical Networks (ICTON)*, Angers, France, pp. 1-5, 2019 doi: 10.1109/ICTON.2019.8840438.
- [58] A. Scire, F. Tropeano, A. Anagnostopoulos, I. Chatzigiannakis, “Fog-Computing-Based Heartbeat Detection and Arrhythmia Classification Using Machine Learning”, *Algorithms*, 12, 32, 2019.
- [59] H. J. de Moura Costa, C. A. da Costa, R. da Rosa Righi et al., *Fog computing in health: A systematic literature review*. *Health Technol.*, 2020. <https://doi.org/10.1007/s12553-020-00431-8>.